

Chapter 3: Getting started

Before you use your 5250 Gateway product, you must complete these tasks:

- Install the 5250 communication adapter board. See the documentation accompanying your host connectivity hardware for installation information.
- Create the action and screen templates necessary to navigate through the host application and return the desired information (discussed in Chapter 2).
- Create the trs.conf and trs.sdf files (discussed in this chapter).
- Download the communication board configurations. You must download the board configurations each time you power up your application processor. You can configure your application processor to automatically download the configurations (this process is described later in this chapter).
- If you are going to connect to the host via modem, make sure you configure the modem with the correct settings and attach it to the 5250 board.
- Start Meridian IVR 2.0/I on the application processor. The TRS processes start when you issue the MIVR start command.
- If you have a dial-up connection, you must know the telephone number of the 5250 host.
- Start the 5250 Gateway.

During startup, the `trs.log` file remains empty unless the verbose option (`-v`) is added. To debug the application put the TRS process into debug mode. Check the `trs.log` file or the `trs.temp` file for errors. To put the TRS process into verbose mode go to `/u/ivr/startup` and change the `.trs -b` line so that it reads `.trs -b -v`.

Note: Everytime any of the control files are modified, (`trs.sdf`, `trs.conf`, action templates, or screen templates) Meridian IVR must be stopped and then restarted so that the `trs` process can pick up the changes.

To check your template and action file run `.trs -c`. This displays information on the version of `trs` and also verifies the correctness of the application templates and action files.

If you need to use the application processor as a terminal connection to the 5250 host, you must first:

- Start the terminal connection software (described later in this chapter).
- Make sure the terminal you use for terminal connection has been previously defined for this purpose. See the documentation included with your communication board for more information.

Setting up the `trs.conf` file

The ***trs.conf*** file specifies the initial-action template for each session you define on the Meridian IVR 2.0/I application processor. The ***trs.conf*** file must reside in the `/u/ivr/3270` directory. The TRS only reads from the `/u/ivr/3270` directory. 5250 sessions, therefore, should be set up under the `/u/ivr/3270` directory. [Figure 3-1](#) shows the syntax for the ***trs.conf*** file.

Figure 3-1
trs.conf file syntax

```
app-name:board-number>session-number:initial-template:heartbeat:protocol
```

The colons (:) and the greater than (>) symbols are used as field delimiters and must be placed in the specified positions without any additional white space.

app-name

The app-name entry identifies the application on the mainframe to which you assign the session. The name entered here is the same name you enter in the action templates that are executed by the Meridian IVR 2.0/I application.

board-number

The board-number entry should be 0 for the first (or only) board, 1 for the second.

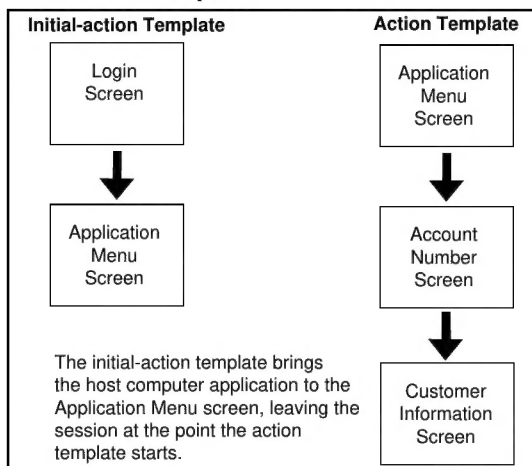
session-number

The session-number entry determines the number of Logical Units to activate at startup.

initial-template

The initial-template entry identifies an initial-action template (without the .act file extension) for setting the startup action for the specified sessions when connecting to the host computer. The start-up action brings the specified sessions to the screen on the host computer where the action templates start when processing requests. **Figure 3-2** shows the flow of a sample initial-action template.

Figure 3-2
Initial action template



The initial-action template follows the same format described earlier for action templates. The screen templates specified by the initial-action template must also be created. The screen template for a login screen (or the login prompt) must specify the correct login ID and password (if appropriate) to access the host computer. You should not specify a reset-action template in the initial-action template, although you should specify a logout-action template.

heartbeat

You can specify an optional heartbeat action for the application specified by app-name. You can use this feature to send an indication to the host that a session is still active. Some hosts log out sessions that remain idle for a period of time. You can also use the heartbeat action to check connectivity, verifying that the session remains on the appropriate screen. Typically, the heartbeat action contains a logout-action template which brings the session back to the login screen if the connectivity with the mainframe fails. If you do not need a heartbeat action, enter a hyphen for this action.

You specify the heartbeat in this format: actiontemplate@n, where n is the number of seconds between each execution of the action template when the session is idle.

The heartbeat action template uses the same syntax as the action templates described earlier in this chapter, and usually only specifies a single screen template. That screen template is usually the last screen specified in the initial-action template. Typically, your screen template would only include a key-descriptor line, usually the ENTER key and validation-tag.

protocol

The protocol entry indicates the communications protocol being used by app-name. This field is required and should be set to 5250.

Example “trs.conf” file

Consider the following example. Initial-action template files login.act and signin.act have been defined and reside in the /u/ivr/3270 directory. There are four applications which you want to access on the host computer:

- accounting, accesses the accounting software
- market, tracks stock market activity
- banking, retrieves credit balances
- airline, for purchasing tickets on a local commuter carrier

The application names shown here are not necessarily the actual names assigned on the host computer, but they are the names assigned on the application processor for the trs.conf file and all action templates that use these applications.

You want to set up the 5250 Gateway to initialize the sessions as follows:

- initialize sessions 2-3 for accounting with signin.act
- initialize sessions 4-8 for market with login.act
- initialize sessions 9-10 for banking with login.act
- initialize sessions 15-17 for airline with signin.act

Figure 3-3 illustrates how you should set up trs.conf to implement this configuration.

Figure 3-3
trs.conf file sample

```
#trs.conf file to set up sessions for the Meridian IVR application processor
accounting:0>2-3:signin:ping@60:3270
market:0>4-8:login:-:5250
banking:0>9-10:login:-:5250
airline:0>15-17:signin:ping@60:5250
```

Setting up the trs.sdf file

The trs.sdf file maps the short name of each session to the LU number. To find the Session User Name and the Session Short Name, run the utility `<express -c -u express>` from the root account. Then, go to “Server Configuration”, and “Define User/Sessions”, and “<Users>”. The Express Username is listed to the right of the User label. The Session Short Name is listed to the right of the Session label. The first LU number must start with 2, then 3 and so on. Addresses 0 (zero) and 1 are reserved.

TRS requires the trs.sdf file when using Apertus's Express host connectivity package. See the documentation accompanying the Apertus Express package for information on configuring Express for your environment. Configuring Express provides you with the values for the trs.sdf file. The trs.sdf file follows this format:

U: <i>Express Username</i>	
Session Short Name	LU Number
Session Short Name	LU Number
Session Short Name	LU Number
•	•
•	•
•	•

You can list up to eight sessions and LUs under each Express Username. You can add comments anywhere in this file by preceding them with a pound sign. **Figure 3-4** shows an example trs.sdf file.

Figure 3-4
trs.sdf file sample

```
U: express
a 2
b 3
c 4
d 5
e 6
f 7
g 8
h 9
U: expressa
i 10
j 11
k 12
l 13
m 14
n 15
o 16
p 17
U: expressb
q 18
r 19
s 20
t 21
```

ATTENTION!

You must add a space in the U: express lines for the code to work.

A complete sample transaction

The following sections provide an example of action and screen templates and their corresponding mainframe screens:

- initial-action template
- action template
- reset-action template
- logout-action template
- screen templates that support the above action templates

The sample transaction being performed in this section uses these templates in the following way:

- The initial-action template logs on to the host computer and starts up the “acct” application.
- The action template chooses the “Accounts Receivable” option from the application’s menu and retrieves a customer’s balance.
- The reset-action template returns to the application’s menu screen and waits for the next transaction.

The logout-action template is included in case an error occurs at any time during the transaction.

Initial-action template

Once the TRS process is invoked, the initial-action template `acctinit.act` brings the session assigned to the accounting application to the menu screen. **Figure 3-5** shows the initial-action template, its supporting screen templates and the corresponding screens on the host computer.

Figure 3-5
Initial-action template sample

Templates	Corresponding Screens
<pre>#initial action template to start #the accounting application #filename: acctinit.act acctinit accounting - clrinit acctlog1 acctlog2 atactmenu</pre>	<pre>login:</pre>
<pre>#screen template that logs #into the host #filename: acctlog1.scn acctlog1 1,1 login: 0,0 - vad 0,0 > ENTER 0,0 - quality 0,0 > ENTER 0,0 @ 3</pre>	<pre>login: vad Password: ***** login: vad Password: ***** Last login: Fri Oct 7 16:01:34 from Publisher ULTRIX V4.2 (Rev 96) System #9: Wed Oct 12 10:08:24 EDT 1994 \$</pre>
<pre>#screen template that starts #the accounting application #filename: acctlog2 acctlog2 0,0 ULTRIX 0,0 - acct 0,0 > ENTER 0,0 @ 5</pre>	<pre>login: vad Password: ***** Last login: Fri Oct 7 16:01:34 from Publisher ULTRIX V4.2 (Rev 96) System #9: Wed Oct 12 10:08:24 EDT 1994 \$ acct</pre>
<pre>#screen template that validates #the accounting menu screen #filename: atactmenu.scn atactmenu 1,20 ACME Accounting</pre>	<pre>ACME Accounting 1. Accounts Receivable 2. Accounts Payable 3. Reports 4. Inventory 5. Exit Enter menu selection:█</pre>

The initial-action template starts the accounting session with the host computer by accessing accounting from the trs.conf file. It then executes the screen templates acctlog1, acctlog2, then atactmenu.

The screen template acctlog1.scn does the following:

- enters the login name
- enters the password
- waits three seconds for the host to process the login action

The screen template acctlog2.scn does the following:

- at the system prompt, enters the command to run the application
- types the ENTER key to start the application
- waits five seconds for the host to run the accounting application

The screen template atacctmenu.scn validates the accounting main menu screen, but performs no action.

The acctlog1 and acctlog2 screen templates use “0,0” as the row,column location and a hyphen for the field-tag to indicate that all text will be entered at the current cursor position. This method is used because the login prompt and the first prompt may not always appear in the same location on the screen. Prompts vary from system to system, and can include longer, customized names.

Note: No key-descriptor is specified for the “atactmenu” screen template. This means once the screen is validated, this screen remains active until the next transaction is executed.

The initial-action template shown in [Figure 3-5](#) specifies the clrint.act logout-action template, and does not specify a reset-action template. [Figure 3-6](#) shows the clrint logout-action template, its corresponding screen templates, and the associated application screens.

Figure 3-6
Logout-action used by the initial-action template

Templates	Corresponding Screens
<pre>#logout-action template for the #acctinit initial-action template #filename: clrinit.act clrinit accounting — — clrmenu logout</pre>	<pre>ACME Accounting 1 Accounts Receivable 2 Accounts Payable 3 Reports 4 Inventory 5 Exit Enter menu selection: █</pre>
<pre>#screen template to exit the #accounting application #filename: clrmenu.scn clrmenu 1,20 ACME Accounting 0,0 — 5 0,0 > ENTER 0,0 — Y 0,0 > ENTER</pre>	<pre>ACME Accounting 1 Accounts Receivable 2 Accounts Payable 3 Reports 4 Inventory 5 Exit Enter menu selection: 5 Do you really want to exit? Y</pre>
<pre>#screen template to log off the host #computer, preparing for the initial #action template to be executed #filename: logout.scn logout 0,0 — 0,0 — logout 0,0 > ENTER</pre>	<pre>\$ logout login:</pre>

The final screen template listed, logout, brings the host application back to the initial screen, leaving the connection open and waiting for the TRS process to log in. TRS always executes the initial-action template for a session (after a pause of 30 seconds) after any logout-action template is executed.

Action template performing a transaction

As shown in **Figure 3-5**, the initial-action template brings the session to the application's menu screen. The action template created to get a customer's balance starts at that screen. **Figure 3-7** shows the action template, its screen templates, and the corresponding host computer screens that perform the following functions:

- chooses the Accounts Receivable option from the application's menu
- enters (on the line that pops up at the bottom of the screen) the customer account number from an input buffer provided by the COMI cell
- when the customer information screen displays, the template places the current balance in an output buffer to be transmitted to the COMO cell
- executes a reset-action template to return the session application to the application's menu screen

Figure 3-7
Action template sample

Templates	Corresponding Screens
<pre>#action template to perform steps #required to retrieve customer's balance #filename: getbalance.act getbalance accounting clr_cust logout_cust accrec #choose Ac. Rec menu acctno #enters account number customer #retrieves balance</pre>	<pre>ACME Accounting 1 Accounts Receivable 2 Accounts Payable 3 Reports 4 Inventory 5 Exit Enter menu selection: █</pre>
<pre>#screen template to choose accounts #receivable option, filename: accrec.scn accrec 1,20 ACME Accounting 0,0 — 1 0,0 > ENTER</pre>	<pre>ACME Accounting 1 Accounts Receivable 2 Accounts Payable 3 Reports 4 Inventory 5 Exit Enter menu selection: 1 Enter account number: █</pre>
<pre>#screen template to enter acct number #in popup field, filename: acctno.scn acctno 1,11 Enter account number: 0,0 — * 0,0 > ENTER</pre>	<pre>ACME Accounting 1 Accounts Receivable 2 Accounts Payable 3 Reports 4 Inventory 5 Exit Enter menu selection: 1 Enter account number: 845-23-87</pre>
<pre>#screen template to obtain balance #filename: customer.scn customer 1,1 Account Number: 2,48 — \$9</pre>	<pre>Account Number: 845-23-87 Customer: Jane K. Smith Current Balance: 2486.14 Address: 19 Alpha Road Payment Due: 150.00 Chelmsford Payment Due Date: 4/30/93 MA 01824 Options: 1 Print invoice 2 Enter payment 3 Enter purchase 4 Exit Enter menu selection: █</pre>

The number 1 and the account number are entered at the current cursor location. Therefore, you do not need to specify a location or a field-tag. The last line in the “customer” screen template places the contents of the field starting at location 2,48 (the number 2486.14) into the next output buffer.

The reset-action template is designed to return the session to the “acct” application’s menu screen where it awaits the next transaction. [Figure 3-8](#) shows the reset-action template, and its accompanying screen templates.

Figure 3-8
Reset-action template sample

Action Template	Screen Template
<pre>#reset-action template for the getbalance #action template, filename: clr_cust.act clr_cust accounting — — clrcust #exits customer info screen atacctmenu #validates application menu screen</pre>	<pre>#screen template to clear customer info #screen, filename: clrcust.scn clrcust 1,1 Account Number: 0,0 — 4 0,0 > ENTER</pre>
	<pre>#screen template that validates the acctg #menu screen, filename: atacctmenu.scn atacctmenu 1,20 ACME Accounting</pre>

Notice that the atacctmenu screen used in the reset-action template is the same screen used in the initial-action template. This screen template verifies that the menu screen has returned, but it performs no function.

Figure 3-9 shows the logout-action template, and its accompanying screen templates, that return the host computer to the login screen if an error occurs and the reset-action template also experiences an error.

Figure 3-9
Logout-action template sample

Action Template	Screen Template
<pre>#logout-action template for the getbalance #action template, filename: logout_cust.act logout_cust accounting — — clrcust #exits customer info screen clrmnu #exits applicatoin menu logout #logouts out to prepare for login</pre>	<pre>#screen template to clear customer info #screen, filename: clrcust.scn clrcust 1,1 Account Number: 0,0 — 4 0,0 > ENTER</pre>
	<pre>#screen template to exit the #accounting application #filename: clrmnu.scn clrmnu 1,20 ACME Accounting 0,0 — 5 0,0 > ENTER 0,0 — Y 0,0 > ENTER</pre>
	<pre>#screen template to log off the host #computer, preparing for the initial #action template to be executed #filename: logout.scn logout 0,0 — 0,0 — logout 0,0 > ENTER</pre>

The logout-action template brings the host computer back to the login screen. The TRS then executes the initial-action template, as it does whenever any logout-action template is successful.

If the transaction fails before the customer screen is accessed, the logout-action template searches the other screen templates until a valid tag match is found, then it executes that screen template and the following screen templates.